<Company Name>

<Project Name>
Software Requirements Specification
For <Subsystem or Feature>

Version <1.0>

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>

Table of Contents

Software Requirements Specification

1. Introduction

*[The introduction of the **Software Requirements Specification (SRS)** provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the **SRS**.]*

*[Note: The **SRS** captures the complete software requirements for the system, or a portion of the system. Following is a typical **SRS** outline for a project **using use-case modeling**. This artifact consists of a package containing use cases of the use-case model and applicable Supplementary Specifications and other supporting information. For a template of an **SRS not** using use-case modeling, which captures all requirements in a single document, with applicable sections inserted from the Supplementary Specifications (which would no longer be needed), see the file titled rup_srs.dot.]*

*[Many different arrangements of an **SRS** are possible. Refer to [IEEE93] for further elaboration of these explanations, as well as other options for an **SRS** organization.]*

1.1 Purpose

*[Specify the purpose of this **Software Requirements Specification**. The **SRS** fully describes the external behavior of the application or subsystem identified. It also describes nonfunctional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.]*

1.2 Scope

*[A brief description of the software application that the **Software Requirements Specification** applies to, the feature or other subsystem grouping, what Use-case model(s) it is associated with, and anything else that is affected or influenced by this document.]*

1.3 Definitions, Acronyms, and Abbreviations

*[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the **Software Requirements Specification**. This information may be provided by reference to the project's Glossary.]*

1.4 References

*[This subsection provides a complete list of all documents referenced elsewhere in the **Software Requirements Specification**. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]*

1.5 Overview

*[This subsection describes what the rest of the **Software Requirements Specification** contains and explains how the document is organized.]*

2. Overall Description

*[This section of the **Software Requirements Specification** describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as product perspective, product functions, user characteristics, constraints, assumptions and dependencies, and requirements subsets.]*

2.1 Use-Case Model Survey

[If using use-case modeling, this section contains an overview of the use-case model or the subset of the use-case model that is applicable for this subsystem or feature. This includes a list of names and brief descriptions of all use cases and actors, along with applicable diagrams and relationships. Refer to the

Use-Case-Model Survey Report, which may be used as an enclosure at this point.]

2.2 Assumptions and Dependencies

[This section describes any key technical feasibility, subsystem or component availability, or other project related assumptions on which the viability of the software described by this Software Requirements Specification may be based.]

3. Specific Requirements

[This section of the Software Requirements Specification contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements and testers to test that the system satisfies those requirements. When using use-case modeling, these requirements are captured in the use cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section.]

3.1 Use-Case Reports

[In use-case modeling, the use cases often define the majority of the functional requirements of the system, along with some non-functional requirements. For each use case in the above use-case model, or subset thereof, refer to, or enclose, the use-case report in this section. Make sure that each requirement is clearly labeled.]

3.2 Supplementary Requirements

[Supplementary Specifications capture requirements that are not included in the use cases. The specific requirements from the Supplementary Specifications, which are applicable to this subsystem or feature, should be included here and refined to the necessary level of detail to describe this subsystem or feature. These may be captured directly in this document or referred to as separate Supplementary Specifications, which may be used as an enclosure at this point. Make sure that each requirement is clearly labeled.]

4. Supporting Information

[The supporting information makes the Software Requirements Specification easier to use. It includes:

- *Table of Contents*
- *Index*
- *Appendices*

 These may include use-case storyboards or user-interface prototypes. When appendices are included, the Software Requirements Specification should explicitly state whether or not the appendices are to be considered part of the requirements.]